LA-UR-20-21328
Approved for public release; distribution is unlimited.

Title: PAGOSA Mesh Block Calculator

Author(s): Wooten, Hasani Omar

Intended for: Report

Issued: 2020-02-12

PAGOSA Mesh Block Calculator

H. Omar Wooten
hasani@lanl.gov[1]

[1]X-Theoretical Design: Primary Physics (XTD-PRI)

February 4, 2020

Abstract

A method for automatically generating the "mesh namelist block" portion of a PAGOSA input file, which specifies the boundaries and zone sizes of the computational mesh, has been developed. This application standardizes the mesh block for 2-, and 3-dimensional problems and is computed after the user specifies the extent and resolution of the mesh and the number of processors which will be used to run the problem. The inputs may be provided directly through a series of command-line guided questions, or a separate input file. The mesh block is written to both standard output and to disk.

1 Introduction

The Los Alamos National Laboratory code PAGOSA [1, 2] solves computational fluid dynamics problems in 2-, or 3-dimensions described on a rectilinear grid with Eulerian algorithms (i.e., material flows and thermodynamic quantities advect through a regular grid that remains fixed in space). In addition to its advanced hydrodynamics algorithms and its speed, setting up a problem from scratch in PAGOSA much easier than other hydrodynamics code packages. This makes PAGOSA an attractive alternative for users of other codes to quickly gain a basic understanding of a specific problem very quickly.

For first time users, perhaps the trickiest part of composing a PAGOSA input deck is writing the "mesh namelist block," which specifies of the rectilinear Eulerian grid spatial domain and resolution, and the number of processors with which the problem will be solved. Writing the mesh block is not difficult, and with practice, it can be accomplished rather quickly. However, for users performing scoping studies or who wish to adapt the calculation to whatever computing resources are available, freqently rewriting the mesh block is inefficient and can potentially extend the time for quick turnaround calculations.

This report describes a method that has been developed to automatically write the entire PAGOSA mesh block following user entry of a series of simple inputs describing the spatial domain of the mesh and the number of processors.

2 The mesh block

The PAGOSA Input Reference Manual [2] provides a complete description of the PAGOSA Eulerian mesh and the mesh namelist block. In summary, specification of the mesh requires the user specify its boundaries in 1-, 2-, or 3-dimensions (separated into sub-segments as desired), the number of cells within each segment (that is, the segment's width divided by the desired zone size), and any geometric ratio multipliers to be applied per segment in each direction. In addition, the user must specify the number of computing processors to be applied to the problem in each dimension. Furthermore, the number of cells and the number of processors in each direction are subject to the following contraints,

$$n_i = \sum_{seg} ncell_{i,seg} + 2 = m_i \cdot npes_i, \tag{1}$$

$$m_i \geq 3, \text{ and} \tag{2}$$

$$\prod_i npes_i = \# \text{ processors}, \tag{3}$$

where:

i is a direction, x, y, or z,
n_i is the number of cells in direction i,
seg is a segment in i,
$ncell_{i,seg}$ is the number of cells in a segment in the i^{th} direction,
m_i is an integer number of cells in the computational submesh ($32 \leq m_i \leq 256$ recommended),
$npes_i$ is the number of processors assigned to direction i, and
$\#$ processors is the number of compuational processors assigned to the simulation run.

3 The calculator

To automate the process of writing the mesh namespace block for changing conditions (either different zone size, or using different computational processors), the PAGOSA Mesh Block Calculator has been developed. The calculator is a script written in Python [3] that collects 4 inputs from the user, either from a command-line interface or from a short ASCII text input file, that includes:

1. number of processors to be used for the simulation,
2. dimensionality of the problem,
3. minimum and maximum limits of the mesh in all directions, and
4. zone size for all directions

Using these inputs, the calculator then:

— for each direction and segment, computes the number of zones, $ncell_{i,seg}$, given the specified zone sizes,

— computes the n_i, m_i, and $npes_i$ required that satisfies the constraints in Equations [1-3],

— recomputes the mesh boundaries given calculations of n_i, m_i, and $npes_i$, and

— constructs the text for the mesh namespace block of the PAGOSA input file, and writes the mesh block both to the screen and to an output file.

4 Examples

This section provides example inputs and outputs for the PAGOSA Mesh Block Calculator, both using the command-line and input file options. The blue text represents user commands and input.

4.1 2-D cylindrical command line

Scenario: A mesh that spans [0.0, 5.0] in the radial (x) direction, and [-10.0, 10.0] in the axial (z) direction, both with zone size of 500 μm.

Input:

```
$ python pagosamesh.py
Enter # processors (integer):  108
1 for 1D
2 for 2D-cartesian
3 for 2D-cylindrical
4 for 3D-cartesian
Enter geometry selection from choices above:  3
Enter x, z, domain(s).  For example:  -5.0 5.0 or -5.0 0.0
5.0
Enter x-domain:  0.0 5.0
Enter z-domain:-10.0 10.0
Enter x, z zone sizes :  0.05 0.05
```

Output:

```
! ======================================================================
! PAGOSA mesh block
! Generated by             : Manual entry
! Processors               : 108
! Zones                    : 44064
```

```
! Resolutions               : x=0.0500, z=0.0500,
! Zones Per Processor       : 408
! =========================================================================
&mesh
  ncellx = 106,
  coordx = 0.0, 5.3,
  ratiox = 1.0,

  ncellz = 203, 203,
  coordz = -10.15, 0.0, 10.15,
  ratioz = 1.0, 1.0,

  npes_x = 9,
  npes_z = 12,
/
```

4.2 2-D cylindrical input file

As an alternative to manually providing inputs through the command line, users can point the Calculator to a simply-formatted input file. A file named `in_2d_cyl.txt` containing the following lines will produce the same output in the example above.

`in_2d_cyl.txt`

```
processors  : 108
geometry    : 3
x           : 0.0  5.0
z           : -10. 10.
zone sizes  : 0.05 0.05
```

To execute:

```
$ python pagosamesh.py -i in_2d_cyl.txt
```

4.3 3-D cartesian command line and input file

Command-line input:

```
$ python pagosamesh.py
Enter # processors (integer):  2520
1 for 1D
2 for 2D-cartesian
3 for 2D-cylindrical
4 for 3D-cartesian
Enter geometry selection from choices above:  4
Enter x, y, z, domain(s).  For example:  -5.0 5.0 or -5.0
0.0 5.0
Enter x-domain:  -10.0 10.0
Enter y-domain:  -10.0 10.0
Enter z-domain:-10.0 10.0
Enter x, z zone sizes :  0.025 0.025 0.025
```

Input file `in_3d_cart.txt`

```
processors  : 2520
geometry    : 4
x           : -10.0 10.0
y           : -10.0 10.0
z           : -10.0 10.0
zone sizes  : 0.025 0.025 0.025
```

Output:

```
! ========================================================================
! PAGOSA mesh block
! Generated by              : in_3d_cart.txt
! Processors                : 2520
! Zones                     : 528806880
! Zone Sizes                : x=0.0250, y=0.0250, z=0.0250,
! Zones Per Processor       : 209844
! ========================================================================
&mesh
  ncellx = 401, 401,
  coordx = -10.025, 0.0, 10.025,
  ratiox = 1.0, 1.0,

  ncelly = 405, 405,
  coordy = -10.125, 0.0, 10.125,
  ratioy = 1.0, 1.0,

  ncellz = 404, 404,
```

```
  coordz = -10.1, 0.0, 10.1,
  ratioz = 1.0, 1.0,

  npes_x = 12,
  npes_y = 14,
  npes_z = 15,
/
```

Modifying the input file `in_3d_cart.txt` by reducing the zone size from 250 μm to 500 μm and reducing the number of processors to 1260 as follows,

```
processors  : 1260
geometry    : 4
x           : -10.0 10.0
y           : -10.0 10.0
z           : -10.0 10.0
zone sizes  : 0.05 0.05 0.05
```

produces the following:

```
! ==========================================================================
! PAGOSA mesh block
! Generated by            : in_3d_cart.txt
! Processors              : 1260
! Zones                   : 65772000
! Resolutions             : x=0.0500, y=0.0500, z=0.0500,
! Zones Per Processor     : 52200
! ==========================================================================
&mesh
  ncellx = 201, 202,
  coordx = -10.05, 0.0, 10.1,
  ratiox = 1.0, 1.0,

  ncelly = 199, 199,
  coordy = -9.95, 0.0, 9.95,
  ratioy = 1.0, 1.0,

  ncellz = 202, 202,
  coordz = -10.1, 0.0, 10.1,
  ratioz = 1.0, 1.0,

  npes_x = 9,
  npes_y = 10,
  npes_z = 14,
```

```
/
```

Note that reducing the resolution required a slight modification of the boundaries in order to satisfy Eqs [1-3], and these new boundaries are reflected in the new mesh block.

5 Discussion

In the above examples, the mesh block is written in a consistent format, one that indicates to the reader some key information about the mesh, including the resolution and the number of processors for which the mesh block is intended. Also demonstrated is ease with which changes in resolution and processors can be applied to produce new mesh blocks.

At present, the calculator recognizes if the minimum and maximum boundaries of a given direction have opposite signs, and if so, ensures that the grid is spaced such that a segment boundaries lies at 0.0. For some users, this might potentially be a limitation. One certain limitation of the calculator is that no geometric ratios are applied in any direction (that is, `ratiox, ratioy, and ratioz` all equal 1.0). For calculations requiring ratio zoning, the calculator is not an appropriate tool, and the user may need to resort to manually writing the mesh block. However, from discussions with staff members in XTD-SS, it seems that ratio zoning is not often used.

6 Conclusion

The PAGSOSA Mesh Block Calculator is an easy-to-use Python script that generates the mesh namespace block of a PAGOSA input deck using command-line or input-file based parameters from the user. The application facilitates creating new PAGOSA input files with different mesh resolutions and available computing resources quickly, thereby increasing efficiency.

The script is available from the LANL XCP-Stash collection of repositories:

```
https://xcp-stash.lanl.gov/users/hasani/repos/pagosamesh/browse
```

The repository may be cloned by using the command:

```
git clone ssh://git@xcp-stash.lanl.gov:7999/~hasani/pagosamesh.git
```

7 Acknowledgements

Thanks to Drs. Andrew Nelson (XTD-SS), Jim Jenkins (XTD-PRI), Brandon Smith (XCP-1), and Von Whitley (XTD-SS) for helpful discussions related to PAGSOA.

References

[1] *PAGOSA Physics Manual: Code Version v17.3.9*, LA-14425-M, Los Alamos National Laboratory, October 2019.

[2] *PAGOSA Input Reference Manual: Code Version v17.3.9*, LA-CP-17-20444, Los Alamos National Laboratory, September 2017.

[3] Anaconda Software Distribution. Computer software. Vers. 2-2.4.0. Anaconda, Nov. 2016. Web. `https://anaconda.com`.